

REMARKS

Claims 1-28 are pending in this application. In section 3 of the Office Action, claims 1, 6-8, 12, 13, 18-20, and 24-28 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Cray, Jr. (U.S. Patent No. 4,128,880, herein referred to as Cray) in view of Chen et al. (U.S. Patent No. 4,661,900, herein referred to as Chen). In section 9 of the Office Action, claims 2-5, 9-11, 14-17, and 21-23 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Cray in view of Chen and further in view of Laudon et al., "Interleaving: a Multithreading Technique Targeting Multiprocessor and Workstations" (herein referred to as Laudon).

I. REJECTIONS BASED ON CRAY AND CHEN

Claims 1, 6-8, 12, 13, 18-20, and 24-28 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Cray in view of Chen. Applicants respectfully submit that even if combined, the combination of Cray and Chen fails to disclose all of the limitations of the rejected claims. For example, claim 1 recites:

1. A programmable processor comprising:
a data path capable of transmitting data;
an external interface operable to receive data from an external source and communicate the received data over the data path;
a register file containing a plurality of registers each having a register width, the register file coupled to the data path and configured to support processing of a plurality of threads and to store a plurality of multiple-bit data elements in partitioned fields, each of the multiple-bit data elements having an elemental width smaller than the register width;
an execution unit coupled to the data path, the execution unit configured to execute a plurality of instruction streams from the plurality of threads, each instruction stream including a single instruction that specifies an arithmetic operation to cause multiple instances of the arithmetic operation to be performed, each instance of the arithmetic operation to be performed using a different one of the plurality of multiple-bit data elements in partitioned fields of at least one of the registers to produce a catenated result; and
wherein each of the multiple-bit data elements has an elemental width, and the data path has a data path width multiple times greater than the elemental width, to allow multiple-bit data elements used for the multiple instances of the arithmetic operation to be transmitted in parallel from the register file to the execution unit, and wherein the execution unit is operable to receive, in parallel, multiple-bit data elements for the multiple instances of the arithmetic operation and execute the multiple instances of the arithmetic instruction to produce the catenated result. (emphasis added)

As the Office Action acknowledges, Cray fails to disclose at least two limitations recited in this claim: (1) a data path allowing multiple-bit data elements used for multiple instances of the arithmetic operation to be transmitted in parallel from the register file to the execution unit, and (2) an execution unit capable of receiving, in parallel, multiple-bit data elements for the multiple instances of the arithmetic operation. *See* Office Action dated March 18, 2009, page 3, last paragraph to page 4, first paragraph (“Cray does not expressly disclose where each of the multiple-bit data elements has an elemental width, and the data path has a data path width multiple times greater than the elemental width, to allow multiple-bit data elements used for the multiple instances of the arithmetic operation to be transmitted in parallel from the register file to the execution unit, and wherein the execution unit is operable to receive, in parallel, multiple-bit data elements for the multiple instances of the arithmetic operation”).

Instead, the Office Action relies on Chen to make up for the deficiencies of Cray. However, Chen also fails to disclose these two claim limitations, as discussed in detail below.

A. Chen fails to disclose a data path allowing multiple-bit data elements used for multiple instances of the arithmetic operation to be transmitted in parallel from the register file to the execution unit

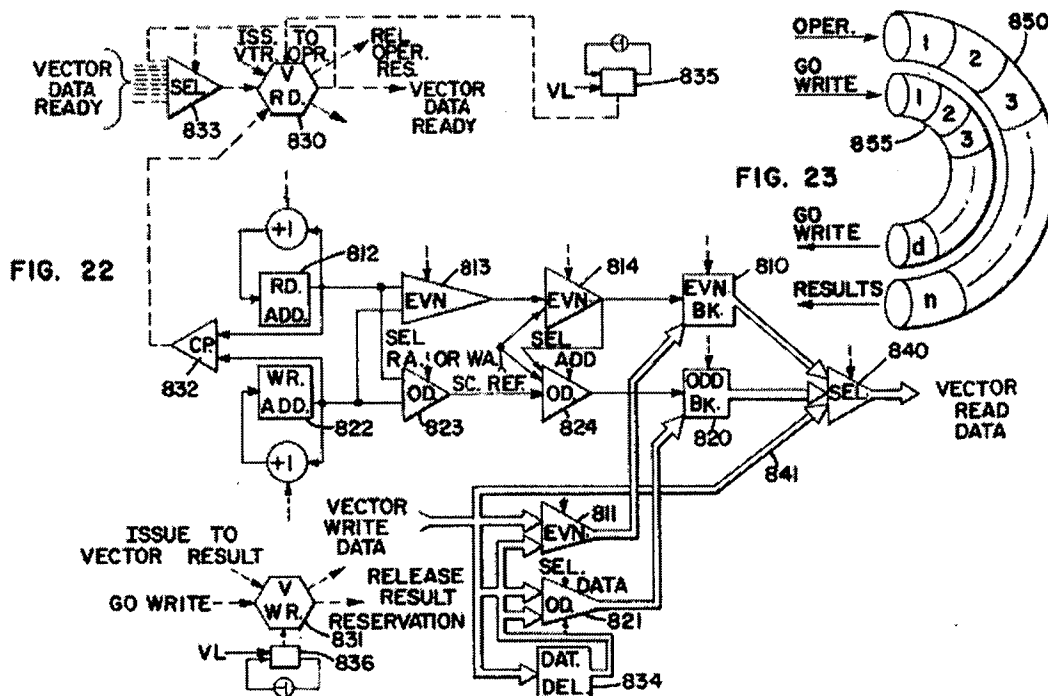
Chen fails to disclose a data path allowing multiple-bit data elements used for multiple instances of the arithmetic operation to be transmitted in parallel from the register file to the execution unit. The Office Action cites two distinct discussions found in Chen, as supposedly disclosing this claimed feature: (1) Chen’s discussion of “even” and “odd” register banks, and (2) Chen’s discussion of accessing operands for a single instance of a vector operation. Neither of these two discussions discloses the claimed feature, as explained below.

1. Chen's discussion of "even" and "odd" register banks fails to disclose the claimed feature

First, the Office Action points to Chen's discussion of "even" and "odd" register banks used to read and write data to and from registers (*See* Office Action dated March 18, 2009, page 4, paragraph 2):

To accomplish this "flexible chaining" capability, the memory circuits of the vector registers, which require one clock cycle to perform a read or write operation, are arranged in two independently addressable banks. One bank holds all even elements of the vector and the other bank holds all odd elements of the vector. Thus, both banks may be referenced independently each clock cycle. Chen, col. 18, lines 38-45.

However, Chen clearly discloses that its "even" and "odd" register banks are never read simultaneously. In fact, Chen's system includes circuitry specifically designed to select either the "even" or the "odd" register bank (not both) as the source of data during a vector read operation. Fig. 22 of Chen is reproduced below (Fig. 22 is presented alongside Fig. 23 in Chen):



As shown in Fig. 22, selector 840 (referred to as "select read data gate 840" in Chen) alternatively selects either the "even" or the "odd" register bank as the data source in a vector

read operation.¹ The existence of selector 840 demonstrates beyond any doubt that only one of the “even” and “odd” register banks can be selected for a vector read operation at any given time. In other words, the contents of the “even” and “odd” register banks are never transmitted in parallel to the execution unit.

The intended use of Chen’s “even” and “odd” register banks is to allow a vector read operation to occur at the same time as a vector write operation. This facilitates flexible chaining of a vector operation. *See* Chen, col. 18, lines 38-45. This type of reading and writing design is also referred to as a “ping pong” arrangement and is well known in the art.² However, it merely teaches that a read and a write operation can be performed simultaneously.

This arrangement does not teach or suggest that two read operations would be performed simultaneously. Indeed, as discussed above, Chen expressly teaches away from reading both the “even” and “odd” register banks at the same time, by disclosing a selector 840 designed to select either the “even” or the “odd” register bank (not both) as the data source in a vector read operation. Thus, Chen’s discussion of “even” and “odd” register banks fails to disclose parallel transmission of data elements used for multiple instances of an arithmetic operation, from the register file to the execution unit, as recited in claim 1.

¹ As a third input, selector 840 can also select the result of the current vector write cycle as the data source for the current read cycle. *See* Chen, col. 20, line 66 to col. 21, line 5. That is, the data source of the vector read operation can be (1) the “even” register bank, (2) the “odd” register bank, or (3) the output of the current write cycle. Selector 840 can only select one of these three inputs as the data source for any particular vector read operation.

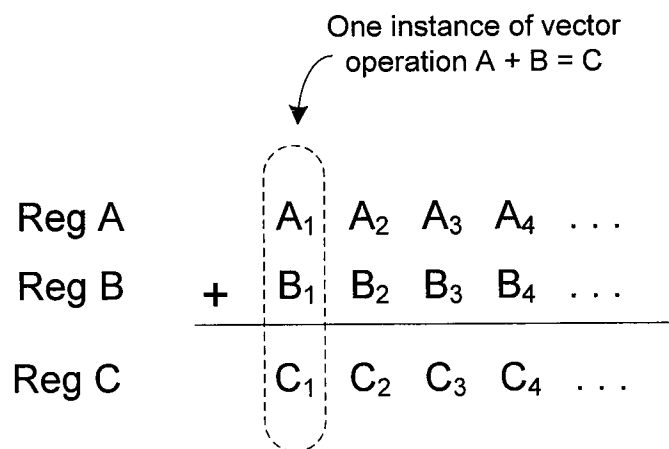
² For example, the elements of a vector may be stored using the “even” and “odd” register banks, such that vector elements 1, 3, 5, 7, etc. are stored in the “odd” bank, and vector elements 2, 4, 6, 8, etc. are stored in the “even” bank. When vector operation 1 is completed, the result of vector operation 1 can be written to the “odd” register bank, and in the same clock cycle, the operand for vector operation 2 can be read from the “even” register bank. When vector operation 2 is completed, the result of vector operation 2 can be written to the “even” register bank, and in the same clock cycle, the operand for vector operation 3 can be read from the “odd” register bank. In this manner, the functional unit can operate more efficiently – *i.e.*, it does not need to wait for the writing of the result of the current instance of the vector operation to be completed before reading the operand for the next instance of the vector operation.

2. Chen's discussion of accessing operands for a single instance of a vector operation fails to disclose the claimed feature

In addition, the Office Action points to another discussion found in Chen, which describes the transmission of different operands used in a single instance of a vector operation (See Office Action at page 4, second paragraph):

In the case where two vector registers are used as operands in a vector operation, each register's read control will monitor the other register's data ready signal to determine when elements are available to be processed by the functional unit. Chen, col. 18, lines 47-51.

This portion of Chen merely refers to the fact that a vector operation may involve two vector operands stored in separate registers. For example, a vector addition may involve adding two vectors operands A and B to produce another vector C. Vectors operand A = [A₁, A₂, A₃, A₄, ...] may be stored in register A, and vector operand B = [B₁, B₂, B₃, B₄, ...] may be stored in register B³:



Here, $A_1 + B_1 = C_1$ is a single instance of the vector operation $A + B = C$. The portion of Chen cited by the Office Action simply notes that the two operands for such a single instance of the vector operation would come from separate registers. For example, A₁ would come from register A, and B₁ would come from register B. Chen also states that each register's read control

³ For example, vector A may be stored in a register having "even" and "odd" banks. Similarly, vector B may be stored in another register also having "even" and "odd" banks.

circuitry would monitor the other register's data signal to determine, for example, when both operands A_1 and B_1 are available to be transmitted to the functional unit. *See* Chen, col. 18, lines 38-45 ("each register's read control will monitor the other register's data ready signal to determine when elements are available to be processed by the functional unit"). In other words, Chen discloses that operands for a single instance of a vector operation (*e.g.*, operands A_1 and B_1) would be made available for transmission to the functional unit at the same time.

However, Chen does not disclose parallel transmission of data elements used for multiple instances of the arithmetic operation, from the register file to the execution unit, as recited in claim 1. Referring again to the picture presented above, for example, Chen does not teach or suggest that operand A_1 (for the first instance of the vector operation) and operand A_2 (for the second instance of the vector operation) can be transmitted in parallel to Chen's functional unit.

In fact, Chen teaches away from such a technique. Fig. 23 of Chen (reproduced above, along with Fig. 22) shows that the function unit processes multiple instances (1 ... n) of a vector operation in a successive fashion. Specifically, instance 1 of the vector operation starts in a first clock cycle, followed by instance 2 in the next clock cycle, followed by instance 3 in the subsequent clock cycle, and so on, until instance n. The multiple instances of the vector operation start one after another, in successive clock cycles. Clearly, the functional unit would receive operands for multiple instances of the vector operation in a sequential manner, not in parallel. Thus, Chen's discussion of accessing operands for a single instance of a vector operation fails to disclose parallel transmission of operands for multiple instances of an operation from the register file to the execution unit.

For the reasons stated above, neither of the two discussions in Chen cited by the Office Action – *i.e.*, (1) Chen's discussion of "even" and "odd" register banks and (2) Chen's discussion

of accessing operands for a single instance of a vector operation – teaches or renders obvious a data path allowing multiple-bit data elements used for multiple instances of the arithmetic operation to be transmitted in parallel from the register file to the execution unit, as recited in claim 1.

B. Chen fails to disclose an execution unit capable of receiving, in parallel, multiple-bit data elements for the multiple instances of the arithmetic operation

Chen also fails to disclose an execution unit capable of receiving, in parallel, multiple-bit data elements for the multiple instances of the arithmetic operation. For this limitation, the Office Action again points to the portion of Chen referring to the fact that multiple operands for a single instance of a vector operation may come from different registers. *See* Office Action dated March 18, 2009, page 4, second paragraph (citing Chen, col. 19, lines 49-51). As discussed above, this portion of Chen merely discloses that operands (*e.g.*, A_1 and B_1 in the picture presented previously) for a single instance of a vector operation would be made available for transmission to the functional unit at the same time. However, Chen does not disclose parallel transmission of data elements used for multiple instances of the arithmetic operation, from the register file to the execution unit.

Indeed, Fig. 23 of Chen shows that the function unit processes multiple instances (1 ... n) of a vector operation in a successive fashion. Specifically, instance 1 of the vector operation starts in a first clock cycle, followed by instance 2 in the next clock cycle, followed by instance 3 in the subsequent clock cycle, and so on, until instance n. In such a system, the functional unit clearly would receive operands for the multiple instances of the vector operation in a sequential manner, not in parallel.

For at least these reasons, Chen not only fails to disclose, but in fact teaches away from, an execution unit capable of receiving, in parallel, multiple-bit data elements for the multiple instances of the arithmetic operation, as recited in claim 1.

The Office Action rejects claims 8, 13, 20, 27, and 28 as being obvious over Cray in view of Chen, based on the same rationale as claim 1. Correspondingly, Applicants traverse these rejections and respectfully submit that claims 8, 13, 20, 27, and 28 are patentable over the combination of Cray and Chen, for similar reasons as stated above with respect to claim 1. Claims 6-7, 12, 18-19, and 24-26 depend from claims 1, 8, 13, and 20, respectively, and therefore incorporate the limitations of their respective base claims. As such, claims 6-7, 12, 18-19, and 24-26 are patentable over the combination of Cray and Chen, for at least the reasons stated above with respect to their respective base claims. Thus, Applicants respectfully request withdrawal of the rejection of 1, 6-8, 12, 13, 18-20, and 24-28.

II. REJECTIONS BASED ON CRAY, CHEN, AND LAUDON

Claims 2-5, 9-11, 14-17, and 21-23 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Cray in view of Chen and further in view of Laudon. Applicants respectfully traverse these rejections for at least two separate reasons, discussed below.

First, even if Cray, Chen, and Laudon were combined in the manner proposed by the Office Action, the resulting combination would fail to disclose all of the limitations of claims 2-5, 9-11, 14-17, and 21-23. As explained previously, the combination of Cray and Chen fails to disclose all of the limitation of claims 1, 8, 13, and 20. Laudon does not make up for these deficiencies in Cray and Chen. Thus, the combination of Cray, Chen, and Laudon still fails to render obvious all of the limitations of claims 1, 8, 13, and 20. Claims 2-5, 9-11, 14-17, and 21-

23 depend from claims 1, 8, 13, and 20, respectively, and incorporate the limitations of their respective base claims. Accordingly, the combination of Cray, Chen, and Laudon would fail to render obvious all of the limitations of claims 2-5, 9-11, 14-17, and 21-23, as well.

Second, Laudon cannot be combined with Cray and Chen in the manner proposed by the Office Action, because Laudon is not prior art to the claimed subject matter. Applicants conceived the claimed subject matter prior to the publication date of Laudon and were diligent up to its constructive reduction to practice, as explained in detail in sections presented below.

A. Laudon is Not Prior Art to the Present Application

Title 37 of the Code of Federal Regulations, section 1.131 provides that:

- (a) When any claim of an application or a patent under reexamination is rejected, the inventor of the subject matter of the rejected claim, the owner of the patent under reexamination, or the party qualified under §§ 1.42, 1.43, or 1.47, may submit an appropriate oath or declaration to establish invention of the subject matter of the rejected claim prior to the effective date of the reference or activity on which the rejection is based...
- (b) The showing of facts shall be such, in character and weight, as to establish reduction to practice prior to the effective date of the reference, or conception of the invention prior to the effective date of the reference coupled with due diligence from prior to said date to a subsequent reduction to practice or to the filing of the application. Original exhibits of drawings or records, or photocopies thereof, must accompany and form part of the affidavit or declaration or their absence must be satisfactorily explained.

Laudon has a publication date of October 1994.⁴ Applicants submit that the claimed subject matter was conceived prior to the publication date of Laudon. Applicants further submit that due diligence was exercised to reduce the claimed subject matter to practice from prior to the publication date of Laudon to the effective filing date of the present application, which represents constructive reduction to practice of the claimed subject matter.⁵

Accordingly, Applicants herewith submit declarations under Rule 131 from Mr. Craig Hansen and Dr. John Moussouris, the inventors of the claimed subject matter. These declarations are submitted along with their accompanying Exhibits A1-A2, B, C6-C17, D1, D23-

⁴ Laudon indicates a copyright date of October, 1994.

D70, and E1-E2, which provide factual evidence of conception prior to October 1994 (publication date of Laudon), as well as factual evidence that the inventors (and their colleagues) exercised due diligence from just prior to October 1994, through August 16, 1995, the date the claimed subject matter was constructively reduced to practice by the filing of U.S. Patent Application Serial number 08/516,036, from which the present application claims priority.

B. Evidence of Conception Prior to October 1994

Conception evidence and activities occurring prior to October 1994 are described in detail in paragraphs 9-12 (pages 8-27) of Mr. Hansen's declaration, and accordingly, will not be repeated here. Exhibits A1 and A2 of the declarations corroborate these activities and events and make clear that the inventors spent considerable time and effort conceiving and documenting their conception. For instance, Mr. Hansen's declaration starting at paragraph 11 includes a table listing each claim element of all pending claims (claims 1-28) and the corresponding evidence of conception as found in Exhibits A1 and A2. The ample amount of evidence presented in the exhibits shows prior conception of all pending claims, including the rejected claims.

C. Evidence of Due Diligence from Just Prior to October 1994, through August 16, 1995

Evidence of due diligence from just prior to October 1994 through August 16, 1995 is described in detail in paragraphs 13-85 of Mr. Hansen's declaration, and accordingly will not be repeated here. Exhibits B, C6-C17, D1, D23-D70, and E1-E2 of the declaration corroborate the relevant activities and events starting just prior to October 1994. As shown by these exhibits and the declaration evidence, the inventors and their colleagues spent considerable time reducing the claimed subject matter to practice up until the constructive reduction to practice of the present

⁵ The Applicants' reliance on constructive, rather than actual, reduction to practice should not be construed as an admission that no actual reduction to practice occurred.

application on August 16, 1995. In particular, the voluminous diligence exhibits show the following activities occurred during the critical period.

MicroUnity retained a team of patent prosecution attorneys from prior to October 1994, through August 16, 1995, and this team of patent prosecution attorneys worked diligently with the inventors to prepare, finalize and file the very detailed '036 patent application, which was filed on August 16, 1995.⁶ The corresponding billing records of the patent prosecution attorneys are presented in Exhibit B to the declarations.

Evidence of MicroUnity's efforts to implement the claimed subject matter in integrated circuit form is shown in email communications among the members of MicroUnity's design team from the time just prior to October 1994, through August 16, 1995. The emails reflect the continual work performed by the MicroUnity design team during this time period to implement the claimed subject matter in integrated circuit form. These emails are grouped and attached as Exhibits C6-C17 to the declarations.

The individuals on the MicroUnity design team spent substantial effort, in the time period from prior to October 1994, through August 16, 1995, to build elaborate databases (sometimes called "tapeouts" or "physical layouts") for the claimed subject matter. Exhibit D1 is a summary of weekly logs of modifications to the electronic databases from prior to October 1994 through August 16, 1995. Exhibits D23-D70 represent actual weekly logs of the modifications made during this time period.

⁶ The case law and MPEP guidance are clear that an applicant can rely on reasonable diligence in preparing and filing a patent application, in combination with reasonable diligence in working toward an actual reduction to practice. *Kondo v Martel*, 223 USPQ 528, 532 (Board of Patent Appeals 1984) ("[A]ctivities directed toward an actual reduction to practice may be considered in conjunction with activities directed toward a constructive reduction to practice in order to show reasonably continuous diligence during the critical period.") (citing *ReyBellet v Engelhardt v Schindler*, 492 F.2d 1380 (C.C.P.A. 1974) for the proposition that "activity directed toward an actual reduction to practice followed by activity by an attorney culminating in the filing of a patent application established the requisite diligence during the critical period"); MPEP 2138.06 ("The diligence of attorney in preparing and filing patent application inures to the benefit of the inventor.").

Exhibits E1 and E2 to the declarations reflect various MicroUnity payroll records from prior to August 1, 2005 through August 16, 1995. Exhibit E1 summarizes monthly total head count of departments at MicroUnity involved in implementing the claimed subject matter in integrated circuit form. Exhibit E2 contains actual payroll records. These exhibits show that MicroUnity spent approximately \$250,000 per monthly pay period from just prior to October 1994 through August 16, 1995, totaling approximately \$3 million dollars of expenditures on payroll for the design team involved in implementing the claimed subject matter in integrated circuit form during the period from just prior to October 1994 to August 16, 1995.

As can be appreciated from inspection of Exhibits B, C6-C17, D1, D23-D70, and E1-E2, the above list of due diligence activities during the critical period is by no means exhaustive, and accordingly, the Examiner is invited to review the detailed evidence in the declaration and the attached exhibits.

The declarations of Mr. Hansen and Dr. Moussouris show conception of at least claims 2-5, 9-11, 14-17, and 21-23 prior to the effective publication date of Laudon as well as the requisite level of due diligence during the critical period. Therefore, Laudon is not prior art to the present application.⁷

Accordingly, Applicants submit that even if Cray, Chen, and Laudon were combined in the manner proposed by the Office Action, the resulting combination would fail to render obvious all of the limitations of claims 2-5, 9-11, 14-17, and 21-23. Furthermore, Laudon is not prior art to the claimed subject matter. As such, Cray, Chen, and Laudon cannot be combined in the manner proposed by the Office Action. For at least these reasons, Applicants request withdrawal of the rejection of claims 2-5, 9-11, 14-17, and 21-23.

⁷Applicants reserve the right to distinguish over the Laudon et al. patent if the PTO does not accept the as persuasive the evidence presented in the Rule 131 declaration.

III. CONCLUSION

In view of the foregoing, Applicants submit that all claims now pending in this Application are in condition for allowance. The issuance of a formal Notice of Allowance at an early date is respectfully requested. If the Examiner believes a telephone conference would expedite prosecution of this application, please telephone the undersigned at telephone number indicated below.

To the extent necessary, a petition for an extension of time under 37 C.F.R. 1.136 is hereby made. Please charge any shortage in fees due in connection with the filing of this paper, including extension of time fees, to Deposit Account 500417 and please credit any excess fees to such deposit account.

Respectfully submitted,

McDERMOTT WILL & EMERY LLP



Eric M. Shelton

Registration No. 57,630

600 13th Street, N.W.
Washington, DC 20005-3096
Phone: 202.756.8000 EMS:MWE
Facsimile: 202.756.8087
Date: September 18, 2009

**Please recognize our Customer No. 20277
as our correspondence address.**